

Release A CDR RID Report

Date Last Modified 1/25/96

Originator Art Gaylord

Phone No (413) 545-2520

Organization Univ. of Mass

E Mail Address art@cs.umass.edu

Document ECS Design Approach

RID ID CDR 3

Review SDPS/CSMS

Originator Ref

Priority 2

Section

Page

Figure Table

Category Name ECS System-Level

Actionee ECS

Sub Category

Subject Evaluation of object oriented design

Description of Problem or Suggestion:

The overall object design of this system seems to be very flat with limited inheritance. As a result there are a large number of objects, many with highly specific operations and attributes. This could make programming more difficult. Some objects are technology or product specific. There is a lack of generic classes from which these special purpose objects are derived.

Originator's Recommendation

Consider hiring an external OOD expert to evaluate this design and if necessary refine the current design.

GSFC Response by:

GSFC Response Date

HAIS Response by: D. Keener

HAIS Schedule 9/5/95

HAIS R. E. R. Meyer/D.

HAIS Response Date 12/8/95

There has been a diligent effort to improve/enhance the ECS Architecture prior to and post CDR. Design issue teams have attacked the key risk areas of the design in such areas as:

- ' Session/Server/Job Model
- ' Subscription Model
- ' Universal Reference Model
- ' Generic Application/Process Framework
- ' OODCE Usage
- ' Error/Event Handling
- ' Logging
- ' Refining/Abstracting Infrastructure

In an effort to increase the project's attention to component level reuse/inheritance, management has put in place a permanent organization composed of strong OO methodologists and project leaders. This organization has also focused on high level architectural abstractions in order to reduce overall risk.

An overall view of the architecture has been the focus, attempting to abstract out commonalities, improve interfaces and validate the design through application of scenarios. However, abstracting common design characteristics into higher level object classes for inheritance requires a careful assessment of the detailed design aspects, otherwise the intended benefits will not be achieved. It is, therefore, a slow process which was not totally successful within the Release A schedule constraints, but also not completely unsuccessful (as evidenced, for example, by the SDSRV object design presented in 305-CD-008-001).

In addition to subsystem-specific activities, design teams were active to extract functionality needed across subsystems into common object classes for re-use across subsystems. Particular care was needed in these areas so as not to drive the re-use effort beyond the point of payback: creating generic classes and abstracting common behavior which is truly re-usable across subsystems is not automatically cheaper if substantial re-use cannot be guaranteed or if the effort needed to research the design is significant.

The activity clearly needs to extend beyond the Release A CDR. Therefore, one major goal has been to make the design extensible in order to support the needs of Release B, and make sure Release B can continue these efforts without serious design breakage. Great strides are being made in these areas.

It should also be noted that ECS Release A is using existing class libraries (e.g., the Hughes Class Library in PLANG and RogueWave classes throughout) which are the model for a large number of low-level object classes (e.g., various types of collections, factories, etc.) which are traditional sources for deep inheritance hierarchies. These hierarchies, thus, would not show up in the ECS detailed design (since the purpose of DID 305 is not to remodel such libraries).

Release A CDR RID Report

show up in the ECS detailed design (since the purpose of DID 305 is not to remodel such libraries).

Status Closed

Date Closed 1/25/96

Sponsor Schroeder

***** Attachment if any *****
